

Testing. Testing Types

SoftServe University November, 2010 Presented October 2008 by Maria Melnyk



Agenda

- What is testing?
- Testing types
- Testing I/O
- Testing types examples
- Testing order
- Testing responsibilities
- Why do we need to test?
- Testing tips





What is Testing?

Software testing is the process of program execution in order to find bugs.

Software testing is the process used to measure the quality of developed computer software. Usually, quality is constrained to such topics as:

correctness, completeness, security;

but can also include more technical requirements such as:

 capability, reliability, efficiency, portability, maintainability, compatibility, usability, etc.





Testing Types: How to choose

- Ensure that the types of testing support the business and technical requirements and are pertinent to the application under test
- Ensure that the activities for each test type and associated phase are included within the master test schedule
- Help identify and plan for the environments and resources that are necessary to prepare for and execute each test type
- Ensure that the types of testing support achievement of test goals



Testing Types: Most common types

- Testability
- Unit Testing
- Integration Testing
- Smoke Testing
- Functional Testing
- GUI Testing
- Usability Testing

- Regression Testing
- Performance Testing
- Load Testing
- Stress Testing
- Acceptance Testing
- Alpha Testing
- Beta Testing





Testability: Definition

Testability is a software characteristic that refers to the ease with which some formal or informal testing criteria can be satisfied. Testability evaluation is started from the **project requirements** analysis phase and lasts up to testing phase start.









Testability: Example

Requirement: While clicking on the "Materials" button the numbered list of books assigned to the selected course should be pulled up in new window. In code Vector template from STL instead of simple array should be used to list books for course.

Testability Note: Limited ability for QA to check whether Vector template is used.

Decision: put the responsibility to test this requirement to developer.



Unit Testing: Definition

Unit testing is a procedure used to validate that individual units of source code are working properly. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.





Unit Testing

 Unit testing – the process of programming, allowing you to test the correctness of the individual modules of the program source.

•The idea - to write tests for each **non-trivial** function or method.

This allows you to:

- Check whether an another code change to errors in the field of tested already the program;
- Easy the detection and elimination of such errors.
- The purpose of unit testing to isolate certain parts of the program and show that individually these pieces are functional.
- This type of testing is usually performed by programmers.
- Unit testing later allows programmers to refactor with confidence that the module still works correctly (regression testing). It encourages programmers to change the code, as is easy to verify that the code works and after the change.



Unit Testing

- It is fashionable to development methodology «TDD» Test Driven Development. The programmer first develops a set of tests for future functionality that calculates all the embodiments, and only then begins to write directly to the production code, suitable for pre-written tests.
- Availability of tests in the program is proof of qualification developer.
- •We work in Visual Studio for Sharpe, and hence the choice is almost limited to two products:
 - Nunit and
 - Unit Testing Framework.
- Unit Testing Framework is a built-in Visual Studio testing system, developed by Microsoft, is constantly evolving.
- Among the latest updates the ability to test UI, and more importantly, it almost certainly will exist as long as there is Visual Studio, which is not true of other tools.



Unit Testing: I/O





Unit Testing: Example

Task: Implement functionality to calculate *speed=distance/time* where distance and time values will be manually entered.

Unit Testing Procedure: Test the implemented functionality with the different *distance* and *time* values.

Defect: Crash when entered value *time=0*.



Integration Testing: Definition

Integration testing is the phase of software testing in which individual software modules are combined and tested as a group. This is testing two or more modules or functions together with the intent of finding interface defects between the modules or functions.





Integration Testing: I/O





Integration Testing: Example

Task: Database scripts, application main code and GUI components were developed by different programmers. There is need to test the possibility to use these 3 parts as one system.



Integration Testing Procedure: Combine 3 parts into one system and verify interfaces (check interaction with database and GUI).

Defect: "Materials" button action returns not list of books but list of available courses.



Smoke Testing: Definition

Smoke testing is done before accepting a build for further testing. It intended to reveal simple failures severe enough to reject a prospective software release. Also is known as Build Verification Test.





Smoke Testing: I/O





Smoke Testing: Example

Task: Test new version of Notepad application.

Smoke Testing Procedure: quickly check the main Notepad features (run application, type text, open file, save file, edit file).

Defect: There is no ability to save file. Button "Save" does nothing.



Functional Testing: Definition

Functional testing is intended to test the application functionality to ensure business logic, required algorithms, use cases flows are met according to the defined functional specification(s).





Functional Testing: I/O

- Passed smoke testing
- Specification
- Test design(s)
- Test case(s)
- Test data
- Testing environment





- testing
- environment



Functional Testing: Example

Task: Test Save feature of Notepad application.

Functional Testing Procedure: test different flows of Save functionality (Save new file, save updated file, test Save As, save to protected folder, save with incorrect name, re-write existed document, cancel saving, etc.)

Defect: While trying to save file with not allowed name (use reserved symbols <, >, :, \setminus , / in file name) no message is shown but Save dialog is closed. File is not saved.

Message "Invalid file name" should be shown and Save dialog should remain opened until correct file name is entered of save process is cancelled.



GUI Testing: Definition

GUI testing is performed to verify the compliance of application GUI with the defined standards. Common standards for some type of applications (Web, Desktop, ...) or specific Client's requirements can be used as basis for testing.









- Standards
- Test design(s)
- Test data
- Testing environment

GUI Testing

- Tested application GUI
- Defect records
- Updated specification, test design(s), test data, testing environment



GUI Testing: Example

Examples of rules:



Defect:

- Any found incompliance with rules is a defect.
- Difference in styles between application windows is a defect.
- Incorrect window position is a defect.

...



Usability Testing: Definition

Usability testing is a means for measuring how well people can use some human-made object (such as a web page, a computer interface, a document, or a device) for its intended purpose, i.e. usability testing measures the usability of the object.

Usability testing generally involves measuring how well test subjects respond in **four areas**:

- time,
- accuracy,
- recall,
- and emotional response.



Usability Testing: Definition

- Time on Task How long does it take people to complete basic tasks? (For example, find something to buy, create a new account, and order the item.)
- Accuracy How many mistakes did people make? (And were they fatal or recoverable with the right information?)
- Recall How much does the person remember afterwards or after periods of non-use?
- Emotional Response How does the person feel about the tasks completed? (Confident? Stressed? Would the user recommend this system to a friend?)



Usability Testing: I/O





Usability Testing: Example

Task: Implement the functionality to view the price of each book in list assigned to some course.

Implementation: "Materials" button action pulls up the "Required Material" dialog with list of books assigned to selected course (name of book and author). "Price" button from "Required Material" dialog pulls up one more dialog with price value for selected book.

Usability Note: Materials are usually being reviewed to at once evaluate how many books are required to learn the course and how much do they cost. It would be more useful if price is shown directly in "Required Materials" dialog near the each item in list in order to avoid one more operation of new dialog opening.



Regression Testing: Definition

Regression testing is any type of software testing which seeks to uncover regression bugs. Regression bugs occur whenever software functionality that previously worked as desired, stops working or no longer works in the same way that was previously planned. Typically regression bugs occur as an unintended consequence of program changes.

Common methods of regression testing include **re-running** previously run **tests** and checking whether previously fixed faults have re-emerged.



Regression Testing: I/O



Testing environment



Regression Testing: Example

Task: Perform regression testing after defect with ability to use reserved symbols in file name while saving in Notepad is fixed.

Regression Testing Procedure: re-run tests to re-test application areas possibly impacted by recent fixes introducing. Set of tests to be re-run is defined basing on impact analysis procedure output. In this particular case Save feature in Notepad should be thoroughly re-tested.

Defect: Any type of defect is possible as soon as any area could be impacted (functionality, GUI, performance, etc.)



Performance Testing: Definition

Performance testing is testing that is performed to determine how fast some aspect of a system performs under a particular workload.

Performance testing can serve different purposes. It can demonstrate that the system meets performance criteria. It can compare two systems to find which performs better. Or it can measure what parts of the system or workload cause the system to perform badly.



Performance Testing: I/O





Performance Testing: Example

Criteria: Server should respond in less than 2 sec when up to 100 users access it concurrently. Server should respond in less than 5 sec when up to 300 users access it concurrently.

Performance Testing Procedure: emulate different amount of requests to server in range (0; 300), for instance, measure time for 10, 50, 100, 240 and 290 concurrent users.

Defect: starting from 200 Concurrent requests respond time is 10-15 seconds.





Load Testing: Definition

Load testing generally refers to the practice of modeling the expected usage of a software program by simulating multiple users accessing the program's services concurrently. Load testing is subjecting a system to a statistically representative (usually) load. The two main reasons for using such loads is in support of software reliability testing and in performance testing.



Load Testing: I/O

- Specification
- Test design(s)
- Test case(s)
- Load test (data, script, etc.)
- Testing environment

Load Testing



 Updated testing inputs



Load Testing: Example

Criteria: Server should allow up to 500 concurrent connections.

Load Testing Procedure: emulate different amount of requests to server close to pick value, for instance, measure time for 400, 450, 500 concurrent users.

Defect: Server returns "Request Time Out" starting from 490 concurrent requests.





Stress Testing: Definition

Stress testing is a form of testing that is used to determine the stability of a given system or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results.

It is subjecting a system to an unreasonable load while denying it the resources (e.g., RAM, disc, maps, interrupts, etc.) needed to process that load. The idea is to stress a system to the breaking point in order to find bugs that will make that break potentially harmful.



Stress Testing: I/O





Stress Testing: Example

Criteria: Server should allow up to 500 concurrent connections.

Stress Testing Procedure: emulate amount of requests to server greater than pick value, for instance, check system behavior for 500, 510, and 550 concurrent users.

Defect: Server crashes starting from 500 concurrent requests and user's **data is lost**.

Data should not be lost even in stress situations. If possible, system crash also should be avoided.



Users (more than 500)



Acceptance Testing: Definition

Acceptance testing is conducted by the end-user, customer, or client to validate whether or not to accept the product. Acceptance testing may be performed as part of the hand-off process between any two phases of development.





Acceptance Testing: I/O





Acceptance Testing: Example

Acceptance Criteria:

- Product should demand all functional requirements listed in requirements specification and change order specifications documents
- User Interface should correspond the requirements mentioned in GUI checklist
- Application should respond in reasonable time assuming adequate hardware backend is provided
- Should not include uncompleted features. All declared features should work as described in SRS documents

Defect: Any type of defect is possible.



Alpha Testing: Definition

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.





Alpha Testing: I/O





Beta Testing: Definition

Beta testing comes after alpha testing. Versions of the software, known as beta versions, are released to a limited audience outside of the company. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.



Beta Testing: I/O









Testing Responsibilities





Testing Environment







Why do we need to test?

If some type of testing is not done, then:

- Testability Testing
- Unit or Integration -Testing
- Smoke Testing
- Usability Testing
- Regression Testing
- Stress Testing

- QA take the responsibility to do impossible work
- QA waste time on useless Smoke testing to find trivial defects
- QA must stop other types of testing after they are partly done
- Users are unsatisfied and product is not competitive
 - Impacted areas stay untested
- Application behavior is not adequate in stress situations



Testing Tips

- The point of testing is to find bugs.
- Bug reports and bug records are your primary work product. This is what people outside of the testing group will most notice and most remember of your work.
- The best tester isn't the one who finds the most bugs or who embarrasses the most programmers. The **best tester** is the one who gets the most bugs fixed.



References

- 1. <u>http://www.defectx.com/</u>
- 2. <u>http://www.sqatester.com/bugsfixes/bugdefecterror.htm</u>
- 3. http://en.wikipedia.org/wiki/Software_testing
- 4. <u>http://en.wikipedia.org/wiki/Functional_testing</u>
- 5. <u>http://www.software-engineer.org/</u>
- 6. <u>http://www.philosophe.com/testing/</u>

Find trainings on: <u>http://portal/Company/Trainings/Forms/AllItems.aspx?RootFolder=%</u> <u>2fCompany%2fTrainings%2fQuality%20Assurance&View=%7bCF8B19</u> <u>C2%2d2E67%2d4CAB%2d97CD%2d01AE58F9B53C%7d</u>



Thank you!

Presented October 2008 by <u>Maria Melnyk</u>